MINIMUM CLASSIFICATION ERROR TRAINING IN EXPONENTIAL LANGUAGE MODELS

Chris Paciorek and Roni Rosenfeld

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
{paciorek,roni}@cmu.edu

ABSTRACT

Minimum Classification Error (MCE) training is difficult to apply to language modeling due to inherent scarcity of training data (N-best lists). However, a whole-sentence exponential language model is particularly suitable for MCE training, because it can use a relatively small number of powerful features to capture global sentential phenomena. We review the model, discuss feature induction, find features in both the Broadcast News and Switchboard domains, and build an MCE-trained model for the latter. Our experiments show that even models with relatively few features are prone to overfitting and are sensitive to initial parameter setting, leading us to examine alternative weight optimization criteria and search algorithms.

1. MCE FOR LANGUAGE MODELING

Language models are typically used in the context of a Bayesian classifier, usually filling the role of the prior, as in speech recognition:

$$s^* = \arg\max_s P(s|a) = \arg\max_s P(a|s) \cdot P(s)$$

where a is the incoming acoustic signal and s is a sentence or utterance.

If the acoustic model structure $P(a|s)$ and the language model structure $P(s)$ are correct, and if there is enough acoustic data and language data to estimate their parameters reliably, then using a maximum likelihood estimate (MLE) for $P(s)$ in the above classifier is guaranteed to be optimal. In practice, of course, none of these assumptions hold. A tempting alternative is thus to set the language model parameters to directly minimize recognition error rate.

This approach, called discriminative training or Minimum Classification Error (MCE) training, has been used in acoustic modeling with some success [1]. But MCE has traditionally been difficult to apply to language modeling, because of the very large number of tunable parameters in a language model, as compared with the modest amounts of utterances available for directly minimizing recognition errors. In fact, the amount of data available for language model MCE training (thousands of utterances) is several orders of magnitude smaller than that available for MLE training (typically in the hundreds of millions of words). In contrast, in acoustic training the same data can be used for either MCE or MLE training.

As a result, the only language model related MCE training in most speech recognition systems is in determining optimal values for the language model weight (and perhaps the word insertion penalty, noise penalty, etc.) from a set of development N-best lists. This is typically done as follows. Let $AS(h) = \log P(A|S = h)$ be the "acoustic score" of a given hypothesis h, let $LS(h) = \log P(S = h)$ be its "language score", and let $N_{\text{wds}}(h)$ and $N_{\text{noise}}(h)$ be the number of words and noise sounds in the hypothesis, respectively. Then the total weighted score for h is:

$$WS_{\overline{\lambda}}(h) = AS(h) + \lambda_1 \cdot LS(h) + \lambda_2 \cdot N_{\text{wds}}(h) + \lambda_3 \cdot N_{\text{noise}}(h) \quad (1)$$

For a given $\overline{\lambda}$, the hypotheses in a given N-best list can be sorted, and the number of errors in the highest scoring hypothesis can be counted. Parameter settings for $\overline{\lambda}$ are then sought to minimize the Word Error Rate (WER) of a given development set of N-best lists. This is typically done using a heuristic grid search such as Powell's algorithm [**?**], which searches the space defined by the λs. Note that the search is not guaranteed to find the global minimum. However, given the low dimensionality of the parameter space (3 in this example), this is not a serious problem in practice. Also, the low dimensionality of the space renders the search computationally feasible, and the small number of parameters avoids overfitting of the model to the development set.

2. MCE WITH WHOLE-SENTENCE EXPONENTIAL LMS

Whole-utterance exponential (a.k.a. maximum entropy) language models [2, 3, 4, 5] are of the form:

$$P(s) = \frac{1}{Z} P_0(s) \exp[\sum_i \lambda_i f_i(s)]$$

where Z is a universal normalizing constant, and $P_0(s)$ is a baseline model, often derived from an ngram, e.g.:

$$P_0(s) = \prod_{i=1}^{|s|} P(w_i | w_{i-n+1}, \ldots, w_{i-1})$$

The $f_i(s)$'s are *features* – arbitrary computable properties of the sentence, and the λ_i are their associated parameters.

In this type of non-conditional model, a relatively small set of features can be designed that capture global aspects of the utterance, such as its grammaticality, semantic coherence, etc. In fact, it can be shown (e.g. [4]) that most of the benefit in such a model is likely to come from common features, namely those that are non-zero a significant fraction of the time (as opposed to, say, individual N-gram features, the majority of which are very rare).

Conditional exponential models are usually trained using the Generalized Iterative Scaling procedure [6] or its variant [7]. This results in the MLE of the exponential family. Whole-utterance exponential models can similarly be MLE-trained using sampling, as was done in [3].

But whole-utterance exponential models can also be MCE trained, as follows. The log-likelihood of a model with k features is given by

$$\log P(s) = -\log Z + \log P_0(s) + \sum_{i=1}^{k} [\lambda_i f_i(s)]$$

The last term is a weighted sum, which can be directly (albeit only locally) optimized for MCE as in equation 1. In fact, the second term ($\log P_0(s)$) can also be assigned a weight, and the other features of equation 1 can be added to the mix as well, for joint optimization.

But the issues raised at the end of section 1 must now be revisited. First, with more than two or three dimensions, a globally optimal (or even near optimal) solution may be harder to find, since there are more opportunities for local minima. Second, the computational requirements of Powell's algorithm may scale badly, making it difficult to do proper model selection (i.e. to try many different sets of features). Last, the increase in the number of parameters means that overfitting to the development set may turn out to be a problem after all.

3. MCE FEATURE INDUCTION METHODOLOGY

In [5] we presented an interactive feature induction methodology for MLE trained exponential models. The base model, $P_0(s)$, is used to generate a corpus of 'pseudo-sentences'. This corpus is compared and contrasted with a corpus of real sentences from the same domain. The goal is for a human observer to detect systematic linguistic and/or statistical differences between the two corpora. When any such difference is encoded as a new feature $f(s)$, its distribution will differ between the two sources, and adding it to the model is guaranteed to result in increased likelihood (i.e., reduced perplexity).

A similar methodology can be used for 'hunting' for features suitable for MCE training. Instead of comparing baseline-generated sentences with real ones, the human observer compares the top-1 decoder hypotheses in a given set of N-best lists with their corresponding transcripts, where the N-best lists were derived using the baseline language model. Again, systematic differences are sought and then encoded in new features $f(s)$.

Another variant on this theme is to compare the true transcripts not only with the top-1 hypotheses, but also with other hypotheses further down in the N-best lists. The underlying observation is that the additional hypotheses contain some examples of recognition errors that are not found in the top-1 hypotheses. These errors can and should be learned from, since they may very well occur in future top-1 recognizer outputs. The additional hypotheses effectively increase the amount of data available for feature induction. We will use a similar argument in section 6 to effectively increase the amount of data for MCE training.

4. FEATURE HUNTING IN BROADCAST NEWS

We began our feature hunting in the Broadcast News domain. Using the methodology described in the previous section, we compared top-1 hypotheses and transcripts corresponding to over 13,000 N-best lists (some 432,000 reference words) derived from the TREC-6 development set by the Sphinx Group at CMU.

An example of a feature that emerged from this methodology is the number of repetitions of pronouns in a sentence. For example if the word "her" appeared twice in a sentence, this was counted as a repetition. Similarly if the words "him" and "he" appeared together in a sentence, this was also counted as a repetition. In a test of the approach, we trained an exponential model *with this single feature* using MCE and Powell's algorithm. We found that this feature alone reduced WER by 0.06% on the training set and 0.05% on a held-out test set (given the number of reference words, these differences are statistically significant).

In searching for more powerful features, we looked at families of related features. One such family we considered was the sequence of verb tenses in a sentence. We focused on the last verb in each verb phrase (e.g. the word "sitting" in the phrase "has been sitting"). This feature can be

Table 1: Ratio of probabilities of verb tense transitions between the top-1 hypotheses and the reference transcripts.

Verb tense	1	2	3	4	5	6
1	1.02	0.93	1.05	1.04	0.84	1.13
2	1.04	0.84	1.06	1.12	0.89	1.16
3	1.16	1.04	0.84	1.03	0.98	0.93
4	1.14	0.94	0.94	0.87	0.93	1.17
5	1.01	0.96	1.06	1.05	0.87	1.07
6	1.18	1.16	0.94	1.31	0.92	0.78

thought of as a long-distance bigram of verb tenses. The six verb tenses we used were 1: base form, 2: non-third person present singular, 3: past participle, 4: third person present singular, 5: present participle, and 6: past tense. First, all the words in each utterance were tagged for part-of-speech We then created a matrix in which every element was the number of sentences in which each tense was followed (possibly at some distance) by some other tense. When we compared these matrices for the reference transcripts and the top-1 hypotheses, we found some large differences for certain pairs of tenses (Table 1). In particular, the frequency with which one tense is followed by the same tense (the matrix diagonal in table 1) is generally lower in the top-1 hypotheses than in the reference transcripts. For example, tense 6 is only 78% as likely to be followed by tense 6 in the top-1 hypotheses as in the reference transcripts. This suggests that the baseline language model used in deriving the N-best lists does not adequately capture same-tense correlations. The small values in column 5 result because this tense is under-represented in the top-1 hypotheses relative to the references — another useful feature.

Other potential features we are considering include same-stem correlations (e.g., bank, banks, banker, banking, banked), the frequency of various parts-of-speech (POS) in a sentence, and other features involving various types of long-distance bigrams, such as bigrams of prepositions or determiners. We would also like to develop a measure of semantic coherence based on the distribution of content words in sentences, then use that measure as a feature. For work in this direction, see [8].

5. APPLYING MCE TRAINING TO SWITCHBOARD

We started our feature hunting in the Broadcast News domain, because we expected the utterances there to be reasonably well formed and correspond roughly to linguistic boundaries, and also because of the large number of N-best lists available in that domain. But, as the hub5 evaluation drew near, we decided to try the approach in the Switchboard domain. To this end we received 1143 development set N-best lists from the SRI hub5 evaluation team.

We first attempted to apply the features we had discovered in the Broadcast News domain to the Switchboard domain. We used a cross-validatory approach to select features for inclusion in the model. We split the training set in half and used MCE and Powell's algorithm to fit the weights on one half of the set and then used these weights on the other half and calculated the resulting change in WER for this held-out set. We examined the following sentence-level features:

- pronoun repetition as described previously
- repetition of words, with the words grouped into 4 classes based on their frequency in the dataset, giving us one feature for the repetition of common words, a second for the repetition of less common words, a third for even less common words and a fourth for the remaining words
- counts of POS for a group of 26 POS from the Penn Treebank tagset
- counts of verb phrases
- counts of noises, laughs, pauses and mouths noises
- count of unknown words (@REJECT@ tokens)
- counts of tense-tense sequences

We found WER reductions in the half-datasets on which the feature weights were estimated. For example, including all 26 POS features, we obtained a WER reduction of 0.6%. Using a reduced subset of 10 POS as well as 15 tense-tense features (selected based on those where the reference transcripts showed the most difference from the top-1 hypotheses), plus the other features mentioned above, resulted in a WER reduction of 1.36%.

However, when we used these weights on the held-out sets, by and large the gains have vanished, suggesting that overfitting is a serious problem even with only several dozen features. We were fitting up to 50 features in a dataset of 573 utterances, resulting in a reduction of up to 100 errors. Our other concern, about the emergence of local minima, was also proven valid: when we ran Powell's algorithm on the same set of features but using multiple different starting values, the results were unstable for some of the features.

Given these negative results, we became more selective about our features, using only a feature or two at a time in each cross-validatory experiment. In particular, we focused on the pronoun repetition feature and the features for same verb-tense repetition. But even in this more selective mode we were unable to significantly improve WER on the held-out set. Whatever improvements we occasionally obtained could not be proven statistically significant due to the small size of the held-ut set.

Thus the features that were useful in the Broadcast News domain were not helpful in the Switchboard domain. This is likely because the Switchboard domain is less structured than the Broadcast News domain and the semantic and syntactic coherence that our features pick up in the latter are much less prominent in the former. In addition, Switchboard utterances tend to be shorter, whereas our features are likely to be more useful in longer utterances where word repetitions and multiple verb phrases are more common. Since we chose our features based on examination of the BN domain, our features may have been ill-suited to the task.

5.1. Conclusions from the experiments

We conclude that local minima, training time and overfitting are all significant obstacles even when the number of dimensions (features) is just a dozen or so. In continuing this work, we plan to focus on constructing fewer but more powerful features, such as the semantic coherence feature mentioned in section 4. This will combat all three problems simulataneously.

In addition, we note that both the feature hunting and feature weighting schemes we employed used only the top-1 hypothesis in each N-best list. As mentioned in section 3, this is suboptimal. The amount of data available for MCE training can be effectively increased by making use of other hypotheses further down the N-best list. We take this up in the next section.

6. ALTERNATIVE WEIGHT OPTIMIZATION CRITERIA

Let $S_{i,j,k}$ represent the feature value, or 'score', of the k'th feature as applied to the j'th hypothesis in the N-best list for utterance i. We will write $\mathbf{S_{i,j}}$ to stand for the score vector ranging over all values of k, and $\mathbf{W}$ for an associated weight vector. Let $WS_{i,j} = \mathbf{W} \cdot \mathbf{S_{i,j}}$ be the total weighted score assigned to hypothesis j of utterance i, and let $NE(i,j)$ be the number of errors in that hypothesis. Then the objective function typically used for N-best rescoring is:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_i NE(i, \arg\min_j WS_{i,j})$$

This function attemps to directly minimize top-1 WER on the development set. To better utilize the other hypothesis in teh N-best list, one might consider some alternative objective functions. For example:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \sum_i RC[NE(i,j), WS_{i,j}]$$

where RC stands for rank correlation. Yet another choice is a rank-weighted error measure:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_i \sum_j [NE(i,j) \cdot f(RANK(WS_{i,j}))]$$

With $f()$ a fast decaying function. Finally one could use error-weighted rank or score:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_i \sum_j [f(NE(i,j)) \cdot RANK(WS_{i,j})]$$

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_i \sum_j [f(NE(i,j)) \cdot (WS_{i,j})]$$

Furthermore, in trying to overcome the computational burden of Powell's algorithm, we have tried to re-cast weight optimization as a regression problem. None of the measurs above are suitable for regression though, because they all involve more than one non-linearlity. For converting into simple regression, Alex Rudnicky had suggested the following: Let $Y_{i,j} = 1$ iff $NE(i,j) = 0$ (i.e. if this is the correct hypothesis), and $Y_{i,j} = 0$ otherwise, then use:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_i \sum_j (Y_{i,j} - WS_{i,j})^2$$

This seems to give up on lots of information though. For now, it seems we must continue to use a grid-search algorithm.

Acknowledgements

We are grateful to Stanley Chen for help with his N-best rescoring tool, and to Andreas Stolcke and the SRI hub5 evaluation team for providing us with their Nbest lists.

7. REFERENCES

[1] Biing-Hwang Juang, Wu Chou, and Chin-Hui Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing, IEEE-SAP*, 5(3):257–265, May 1997.

[2] Ronald Rosenfeld. A whole sentence maximum entropy language model. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997.

[3] Stanley F. Chen and Ronald Rosenfeld. Efficient sampling and feature selection in whole sentence maximum entropy language models. In *ICASSP-99*, Phoenix, Arizona, 1999.

[4] Xiaojin Zhu, Stanley F. Chen, and Ronald Rosenfeld. Linguistic features for whole sentence maximum entropy language models. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, Budapest, Hungary, 1999.

[5] Ronald Rosenfeld, Larry Wasserman, Can Cai, and Xiaojin Zhu. Interactive feature induction and logistic regression for whole sentence exponential language models. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Keystone, CO, December 1999.

[6] J.N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43:1470–1480, 1972.

[7] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 1997.

[8] Can Cai, Larry Wasserman, and Roni Rosenfeld. Exponential language models, logistic regression, and semantic coherence. In *Proceedings of the Speech Transcription Workshop*, 2000.